Social Bots: Human-Like by Means of Human Control?

Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann

{christian.grimme, mike.preuss, lena.adam, trautmann}@uni-muenster.de

University of Münster

Department of Information Systems

Leonardo-Campus 3

48419 Münster, Germany

Abstract

Social bots are currently regarded an influential but also somewhat mysterious factor in public discourse and opinion making. They are considered to be capable of massively distributing propaganda in social and online media and their application is even suspected to be partly responsible for recent election results. Astonishingly, the term 'Social Bot' is not well defined and different scientific disciplines use divergent definitions. This work starts with a balanced definition attempt, before providing an overview of how social bots actually work (taking the example of Twitter) and what their current technical limitations are. Despite recent research progress in Deep Learning and Big Data, there are many activities bots cannot handle well. We then discuss how bot capabilities can be extended and controlled by integrating humans into the process and reason that this is currently the most promising way to go in order to realize effective interactions with other humans.

Social Bots: Human-Like by Means of Human Control?

Contents

Introduction

Social media is a phenomenon that exists for a bit more than a decade now (Facebook went online 2004, Twitter in 2006). For the first time, a large part of the world population is enabled to participate in direct and partly world-wide visible information exchange.

Together with the increasing importance of the social media in all-day live and a growing reach of these networks, their use (or misuse) for orchestrated information distribution in terms of advertisement up to political propaganda becomes attractive for different stakeholders. Due to the underlying technical nature of the communication medium, automated and thus cost efficient access to social media channels is easy. Like for email services several years ago, social media channels are used for simple spamming (Tynan, 2012). However, since about 2010, reports on trolling or automated so-called social bot activity in social media increase - especially with a focus on political manipulation and propaganda (Chu, Gianvecchio, Wang, & Jajodia, 2010; Elliott, 2014; Fredheim, 2013). Today, it is not doubted that social bots have a high societal impact (Cordy, 2017), whatever the approaches realizing them currently are. This leaves research with new and multidisciplinary challenges: Detecting and fighting automated and orchestrated manipulation via social media necessitates insights and understanding of motivation, processes, economics, and current limits of manipulation. Computer scientists track networks, measure interactions, build algorithms, and are concerned with security issues, but are unfamiliar with communication aspects and effects. Social scientists have to understand new (semi-automatic) ways of distributing information or propaganda and answer questions of possible societal impact. Both have to collaborate with statisticians and researchers in the area of artificial intelligence to understand challenges and limits of developing big data-based detection mechanisms.

As a first step, this work covers technical details and processes, economic considerations, as well as limits of automated manipulation via social networks in a multi-disciplinary way and provides some baseline for further discussion. For an initial common understanding, we review the existing interpretations of the term "social bots"

and propose a consolidated definition. This definition is complemented by a comprehensive discussion and classification of automated actors in the web.

Then, we focus on the technical details and challenges in the development of social bots. We first show the construction and implementation principle of a responsive Twitter bot and extend this implementation to a framework for realizing human-like behavior. Additionally, the second concept is validated by a social bot experiment at Twitter, applying 30 of our social bots for gaining followers and distributing (ethically harmless) content. For both concepts we discuss the costs of realization.

In a final step, we address the existing gap of automation on the behavioral level and automation on the communication level. Here, we argue that it is currently most cost efficient to automate bots on a behavioral level, while content generation and bot-human-communication is still steered by humans. In the context of the performed bot network experiment, we empirically show that current automatic detection mechanisms cannot significantly distinguish hybrid bots from human users.

Definition and Taxonomy of Social Bots

When journalists, bloggers, or scientists report on social bots and their potential influence on society, many of these articles provide an own definition of the term "social bot". Very often, these definitions strongly differ from each other, some focusing on technical details, others highlighting social interaction. Sometimes, the definitions even contradict each other or explicitly exclude a class of "social bots" others include. Although the capabilities and effects of social bots taking part in public internet communication are more and more discussed, no common understanding of the vehicle itself has evolved.

As the term itself suggests, definitions stem from a mixed, partly social science and partly technical perspective, while the weighting of the perspectives is usually up to the respective definition's author.

From a technical perspective, the term "bot" is often related to robots, automation and algorithms (Maréchal, 2016). All of these terms are certainly part of the

understanding of social bots, however, their equivalent substitution interweaves technically different concepts like algorithms and robots in a simplistic way and may lead to misunderstandings. Geiger (2016) defines social bots—in a more exact but still very general way—as automated software agents. Emmer (interviewed in Heinrich-Böll-Stiftung (2017)) adds properties like artificial intelligence and the ability to autonomously act in the web.

The social science perspective usually addresses the social or political implications of the actions of social bots. Woolley (2016) states that social bots "mimic human social media users" and "manipulate public opinion and disrupt organizational communication". He also defines so called "political bots" as a special case of social bots. Hegelich (2016) highlights that social bots are hidden actors with a political agenda. He explicitly distinguishes them from "chat bots" or other "assistants". A wider definition which specifically considers the communication behavior is given by Frischlich, Boberg, and Quandt (2017). The authors point out, that the imitation of human communication (behaviour) is a key feature of social bots. This certainly also includes chat bots. Even more general, Kollanyi, Howard, and Woolley (2016) consider interaction with other users through automated social media as key property of social bots. Interestingly, social media platforms like Facebook recently recognized possible effects of social bots by admitting "false amplifications", however, they do not use the term "social bot" throughout their publication (Weedon, Nuland, & Stamos, 2017).

Many application examples of social bots are presented in a recent overview paper by Ferrara, Varol, Davis, Menczer, and Flammini (2016). This work allows to identify many types of bots and check the available definitions. Additionally, the authors give a good but (in our view) slightly too tight definition of social bots: "A social bot is a computer algorithm that automatically produces content and interacts with humans on social media, trying to emulate and possibly alter their behavior." We will keep several aspects of this definition but do not restrict ourself to social media alone. Additionally, we include the communication aspect introduced by Frischlich et al. (2017) and cover the interaction property by referring to agent behavior:

The term "Social Bot" is a superordinate concept which summarizes different types of (semi-) automatic agents. These agents are designed to fulfill a specific purpose by means of one- or many-sided communication in online media.

The most significant difference to other definitions is that we define social bots as a high-level concept which comprises many types of specific bots. Additionally, our definition covers:

- fully automated as well as partly human-steered bot action,
- autonomous action (agent-like),
- an orientation towards a goal,
- multiple modes of communication,
- and a wider ecosystem (all online media).

In the following paragraphs, we give several examples of social bots and specific sub-types as well as of bots which are not covered by our definition and, thus, are not supposed to be social bots.

Social Bots

The most popular type of a social bot is the chat bot, „a software system, which can interact or "chat" with a human user in natural language such as English.“ (Shawar & Atwell, 2007a). The oldest and best known chat bot may be Joseph Weizenbaums ELIZA. It was able to participate in a discussion on psychological topics, controlled by scripts that discover context by identifying keywords (Weizenbaum, 1966). By means of pattern matching, ELIZA answered questions in a very human like manner, so sometimes participants did not even recognize that they talked to a machine instead of a real therapist. The recent wave of chat bot development probably originated in the context of the "Loebner Prize" competition[1], where Hugh Loebner set the task to find the most human-like acting program (Mauldin, 1994).

Nevertheless, chat bots are only as intelligent as their scripts and the databases

[1] http://www.loebner.net/Prizef/loebner-prize.html

behind those scripts are. That is why they are often developed only for specific topics. Nowadays, lots of different chat bots with different aims exist [2]. Meanwhile, companies often use chat bots to handle customer service issues. One can find them "in daily life, such as help desk tools, automatic telephone answering systems, tools to aid in education, business and e-commerce." (Shawar & Atwell, 2007b). As chat bots are created to communicate in dialogs with specific users or customers, a multitude of chat platforms are conceivable, e.g. private chats of social media pages, as well as other online media like email or help sections on private company websites. The bots partly replace human interaction and are often used to do simple preprocessing tasks, e.g., figuring out the right contact person for a specific service issue.

Whereas chat bots focus on one-to-one communication, spam bots are developed to reach a large audience. The goal of this one-to-many communication is spreading information, advertisements or fishing links, without involving the recipient. As they were used to communicate a certain message on behalf of a company, group, or person, they nevertheless fall into the category of social bots.

As mentioned earlier, political bots can be seen as a special type of social bots with the aim to spread political content or participate in political discussions on online platforms (Woolley, 2016). Political bots are designed by politically oriented groups to represent their opinions and mindsets. A typical goal of political bots is, e.g., boosting the popularity of a specific idea or person on a (social) media platform, by generating 'likes' or 'follows'. Furthermore, political bots may make use of the characteristics of chat bots or spam bots. They discover public conversations, posts and comments by identifying keywords and intervene or flood them with own (propagandistic) content. Whether political bots are able to participate in simple conversations with other users or just spread spam in a not reactive way is just a question of the aim (and technical skill) of the operator and the code behind the bot profile. Human-like political bots that act on social media platforms as Twitter and Facebook or comment in forums are potentially capable to influence other users. Especially, if there are many bots

[2] https://www.chatbots.org

cooperating in bot networks, they are able to arise a potentially undeserved awareness to political moods. Examples of potential bot armies were, e.g., discovered in the context of the U.S. presidential election 2016. Bessi and Ferrara (2016) found out that nearly 19 percent of all election-related Twitter posts during this time were made by bots. Furthermore, the German news page "Spiegel Online" reports on parties which considered the usage of social bots supporting their election campains for the German parliamentary elections of 2017 (Pfaffenzeller, 2017; Rosenbach, 2016).

Another type of social bots is the class of mobile phone assistants. Software like Apples Siri[3] is designed to manage human to machine communication with the input of natural language. Nearly any possible functionality of the mobile phone can be used just by voice commands. In this case, the social bot acts as a translator between human users and the phone. With the help of voice recognition and keyword identification, the program figures out appropriate actions or search results for the user.

Bots Not Regarded as Social Bots

Bots that are not covered by our definition of social bots are, e.g., content management bots, aka 'curator bots'. The job of a curator bot is to manage or collect content and to present it in an easy-to-digest way to humans. In contrast to social bots, for curator bots the communication aspect is not pronounced; they only work 'silently' with content. Wikipedia bots are an appropriate example for this class of bots. Pywikibot[4] helps users to nurture articles by deleting superfluous whitespace, generating links to related pages or correcting typos. Another example of content bots are data aggregation bots which are built to manage data and are used for analysis only.

Game bots help their users to be successful in games. Tasks of these bots can be as various as the games they are used in. Game bots can act as opponents in order to enable training, help to navigate through the game or can be used for cheating or just as stand-in for short periods of unavailability (afk). So-called farming bots as e.g. in

[3] https://www.apple.com/de/ios/siri/

[4] https://www.mediawiki.org/wiki/Manual:Pywikibot/Overview

games like World of Warcraft[5] assume simple tasks and free players from time-consuming but necessary duties (Mitterhofer, Kruegel, Kirda, & Platzer, 2009). Nowadays, game bots that realize all these functions and more are available in USB stick format from graphics card vendors. Instead of social bots, game bots focus not on communication and interaction but exclusively on substituting users by imitation.

Service-Level-Agreement (SLA) negotiators focus on machine-to-machine communication. These bots are built to handle Service Level Agreements autonomously. Again, there is no human communication or interaction aspect regarding this class of bots, which is why they are not covered by our definition of social bots.

Discussion

As also shown by the categorization into social and not social bots above, we see the human-machine interaction as a key-factor. Social bots automate social interaction via communication Every online medium where human communication through publicly visible posts, chats, comment-functions, direct messages, etc., takes place, is a possible point of connection for the involvement of social bots. Nevertheless, our definition of social bots should be seen more or less as a high-level concept. Social bots appear as different from each other as the reasons they were built for, and have to be discussed in their specific context. Having a look at the mentioned examples, one can see that there are more and different tasks for social bots than to influence people. Some social bots just substitute their users by assuming duties or handle simple preprocessing tasks. Announcements as Facebook's support for group bots and bot repositories[6] lead us to expect that social bots are going to be a more and more pervasive part of our internet experience in the coming years. When discussing the possibilities of social bots to influence single users up to whole societies, we shall therefore employ more precise notions and terms.

[5] https://www.worldofwarcraft.com/

[6] https://techcrunch.com/2017/03/29/facebook-group-bots/

Automation using Social Bots

The application of social bots for multiple purposes (from advertisement to propaganda) implies different technical challenges as well as economic considerations to be handled. On the one hand, costs are rapidly increasing in terms of technical complexity (e.g. for making social bots more human-like). On the other hand, simple technical realizations may have a big enough impact to maximize monetary or social/political revenue in some cases. In the following sections, we will present a most simple technical realization of a social bot and extend it to a moderately complex behavioral human-like actor on Twitter—trying to keep costs rather low. We then present an experiment using 30 of those Twitter bots and lead over to an economic discussion of hybrid extensions of social bots in the next chapter.

A simple reactive Twitter Bot example

One of the most simple ways to develop a reactive social bot adopts the Twitter Stream API[7]. This basically means, that we listen to the ongoing worldwide Twitter activity and react to arriving posts. More formally, we use a Twitter Stream Listener component that registers with Twitter and additionally implement a simple actuator component which is triggered by incoming Twitter posts and uses the Twitter REST API[8] to reply to these posts, if applicable. Figure 1 provides a schematic overview of the components and data flow of the social bot. For the full implementation details using the Python `tweepy`[9] framework, refer to the listing given in appendix A. Note that due to bandwidth management of the public Twitter Stream, only a subset of posts will reach the listener. Depending on the registered topics and activity at the platform, only about 1% up to 40% (in very restricted cases also more) of the actual traffic may arrive at the listener (Ferrara et al., 2016).

Functionality. The presented social bot enables us to react on twitter posts directly, answering to the sender. In our implementation, the Twitter Stream listener

[7] `https://dev.twitter.com/streaming/overview`

[8] `https://dev.twitter.com/rest/public`

[9] `http://www.tweepy.org/`

consumes the current Twitter Stream with respect to a given set of hash tags or topics. Thereby, we are able to adapt to a specific context or domain of interest. Although the current implementation only greets the user of current post, the functionality of the actuator can easily be extended. The application of this bot ranges from simple demonstration (and greeting) purposes to simple service activities based on standardized responses, e.g.:

- Returning the weather forecast for a city or region mentioned in the current post. Therefore, the actuator may use external weather information sources like OpenWeatherMap[10].
- Answering questions on specific topics detected in the current post. Using the Google Knowledge Graph[11], a mighty ontology network can be connected to the bot, covering an enormous knowledge base.
- In a political context: Respond to specific topics and confront users (usually independent of content posted) with a number of fixed political statements.

Only these three application examples demonstrate the potential of a very simple social bot implementation comprising not more than 30 lines of code for a fully functional frame.

Costs. Obviously, the costs for developing a simple service social bot can almost be neglected. Implementation time is certainly lower than one hour for a medium experienced developer (including error handling code, which is not provided in our listing). The main effort has to be put into the setup for the bot's Twitter account and access to the Twitter API. Therefore, a standard Twitter account must be created and connected to a mobile number for developer access. Both can easily be done in an anonymous way using a fake email address and an invalid or anonymously registered mobile number.

Clearly, the behavior of the presented bot can easily be detected as automated action. No human recipient of a message will consider it to be sent by another human. Instant reaction, permanent activity, and restricted capabilities to analyze content and

[10] `https://openweathermap.org/api`

[11] `https://developers.google.com/knowledge-graph/`

react dynamically will expose the social bot as such.

A Social Bot with human-like behavior

Development of a social bot with sophisticated human-like behavior addresses three main challenges:

1. Producing credible and "intelligent" content, which is accepted as such by human consumers.
2. Leaving a trace of human-like meta-data in social networks.
3. Creating an adequate (often balanced) network of friends or followers to spread information.

While the first challenge is a rather open issue in science and even the more in practice (we will comment on this later), the second aspect can be handled to a certain extent by imitating human actions in social networks sticking to normal human temporal and behavioral patterns. This includes performing activities in a typical day-night-cycle, carefully measured actions at the social media platform, as well as variability in actions and timing. Thus, at Twitter, a bot should pause actions to simulate phases of inactivity (e.g. sleep or work), limit posting and re-tweeting activities to a realistic, human-like level, and vary these pauses and limits.

Another key issue is to grow a network of followers or friends. For social media, a network of friends implies a certain reach: the larger the network of followers, the more Twitter users receive distributed content of the respective account. To create a network, Lehmann (2013) proposes an effective strategy based on a simple observation: users follow other users hoping that those follow back again (which they often do, if the pro-active profile does not obviously look bot-like), thereby establishing a friendship relation. In case this does not happen within a certain time span (i.e. the other user does not follow back), the one-sided connection is often dissolved to keep a balanced following-follower-ratio. An exception to this are very prominent accounts with usually strongly imbalanced following-follower-ratios (far more followers than followed users) or accounts that are mainly used to distribute advertisement (far more followed users than

followers). These clearly indicate not human-like behavior and are used for bot-detection. The overall principle of "follow-for-follow" is not only respected by most human users but can also be applied to grow the follower network of a bot account.

Extending bot functionality. The previously presented simple social bot can easily be extended to fulfill the challenges two and three. Therefore, several Actuators are created that independently perform specific actions on Twitter. Considering the schematic depiction in Figure 2, we briefly describe the important components:

CollectionActuator: This component listens to the Twitter stream and stores user names as candidates to follow later on. The selection of following candidates can be made with respect to different characteristics like the following-follower-ratio (i.e. balanced accounts are preferred), activity on Twitter (potential multiplicators are preferred), Tweet properties (e.g. users sending popular tweets are preferred).

BotProfile: The personal profile of a social bot is defined using a dedicated component which stores all constraints and guidelines to simulate a certain behavior. Here, the day-night-cycle and rest periods can be defined, general parameters for the posting and re-tweeting behavior can be set, and an individual following behavior is formulated. Note that all settings should be guidelines only, in order to add some random variability. The component also provides functionality to request the next action time for all other actuators. This function interprets the given behavior values and (adding some random noise), proposes the next action.

FollowActuator: This actuator ensures a continuous execution and management of the follow-for-follow procedure. With respect to the BotProfile, the component follows a certain amount of previously collected users (see CollectionActuator) and supervises reactions. Is a contacted user follows back, the component adds this user as friend. In case of no response within a certain time window (i.e. 24 hours), the one-sided friendship is canceled and the user is blacklisted.

PostActuator: This Actuator enables the bot to post or re-tweet on Twitter. Therefore, a database of individual tweets and collected tweets is accessed. The amount of

actions is determined by the BotProfile.

PictureActuator: The ability to post pictures on Twitter is implemented by this Actuator. Analog to the behavior of the PostActuator, pictures and matching comments are extracted from a picture database and posted on Twitter.

Experimental evaluation. In order to evaluate our mimicry approach for human behavior in the real-world context, we set up an experiment comprising 30 Twitter bots. In cooperation with the German TV station Pro7, we created 30 fake profiles, see also Table 1, and equipped them with the social bot framework described before. Each bot ran the same code, however, we individualized the bot profiles and Twitter Stream listeners. Each social bot had its own day-night-cycle, activity pattern, and following behavior. Additionally, each bot listened for an individual set of topics within the Twitter Stream. Overall, the experiment was divided into three phases:

1. Build a network of followers over a setup and testing period of 2 days and the following 8 days of combined action. Thus, the experiment lasted 10 days in total of which only the 8 productive days were documented. Note that half of the social bots were mutually befriended by default, while the second half started with no followers at all.

2. Publish content in a coordinated way to test the potential of setting a trend on Twitter. The published content was devised by human actors and only distributed by the bots.

3. Reveal the social bot identity of the respective fake accounts to followers and the public (supported by a TV documentary on the experiment, which is available in German[12]).

Especially phase one demonstrated, that the follow-for-follow approach could successfully be applied to acquire followers automatically. As shown in Figure 3, the amount of followers continuously increased for the evaluated eight days, resulting in

[12] http://www.prosieben.de/tv/galileo/videos/2016347-social-bots-das-experiment-clip

about 1350 followers after this short time period. During the second phase, two (harmless and humorous) hash tags were promoted to test the reach of the acquired followers. Although the hash tag briefly appeared in the German top 100 trending topics, a significant trend could not be established. However, phase three showed, that many human followers had been deceived by the fake bot identities and actions. Reactions from Twitter users were different ranging from disappointment to anger and from amusement to disbelieve.

Although our experiment is only a snap-shot of what is possible by applying human-like acting social bots, some important insights can be extracted:

- Tedious tasks as building a follower network, as well as posting and re-tweeting content, may be automated without being exposed as bot.
- The automatically generated network can be used to spread content to all followers at any point in time. This will cause at least brief visibility and possibly push a topic in order to reach wider popularity.
- Human users can easily be deceived by simple, but fairly realistic social bots behaviors.

Certainly, an important ingredient for the success of our social bots was—besides the human like behavior patterns—the human-generated content published by all bots. As mentioned before, we used manually generated content to be spread by the bots. We include the discussion of this aspect into our cost review.

Costs. The development time of the extended Twitter bot (less than two days) can still be neglected compared to the functionality and benefit of automation provided by the general framework. The more tedious task was to generate all thirty fake accounts on Twitter. Thereafter, we were able to deploy the same code thirty times with only minor adaptations regarding the individual configuration of each bot. Then, phase one (growing the network) was performed by all bots without any human intervention.

Likewise, publishing content in phases two and three needed no intervention. However, content was not automatically generated but provided by humans. We decided

to do so after reasoning on the following to questions: What would have been the costs of generating content automatically, and what quality of content can be achieved?

Implementing the generation of *intelligent* and *creative* content for our hash tags would have cost far more effort than setting up the whole social bot framework. Simple approaches based on templates still require some human interaction and lack creativity. More complex generators based on learned patterns still follow firm rule sets, which limits the variability of linguistic expression. Both probably would have had reduced the credibility of our social bots due to repetitive content. Furthermore, due to the application of thirty cross-linked social bots and their continuous re-tweeting behavior, a single message was repeated many times by other bots and followers, thereby extending its range automatically.

Hybrid Social Bots

The extended social bot framework presented in the previous chapter is able to mimic human behavior on the action level, i.e. a social bot is able to automatically create a follower network, and manage content. Content production, however, is done by human actors. Figure 4 qualitatively shows the automation-orchestration relationship of human users and simple social bots as single actors and as human troll farm or "bot army" respectively. Hybrid social bots are, with respect to automation, an intermediate class of fully automated (behavioral simple) bots and purely human users. Used under orchestration, communication approaches and activity patterns of single actors differ: The army of simple bots is often following a mere client server model with rather similar acting single bots and little autonomy per agent. Hybrid bot networks are certainly still centrally controlled. Each bot, however, possesses a behavioral autonomy, which mimics human behavior. In contrast, Human troll farms are acting on a central interest, context or overall goal but have the highest autonomy per agent. For them, a central content generation becomes dispensable.

In this chapter, we argue that hybridization of bots is an effective (compared to an army of social bots) and low-cost (compared to a human troll farm) approach to gain a

high potential of influence via social media by simulating human behavior and speech. We will show that a network of these hybrid bots is able to sufficiently outsmart current automatic detection mechanisms as BotOrNot[13] (Varol, Ferrara, Davis, Menczer, & Flammini, 2017).

Hybridization as low-cost mimicry approach

The current societal opinion on bot technology seems to be driven by recent success stories of AI, for example the prominently featured wins of an AI against world-class Go players. These successes follow to a large part from the development of deep learning algorithms (Lecun, Bengio, & Hinton, 2015) that are a) able to employ big data collections for learning and b) benefit from extreme parallelization. However, at the core of deep learning successes, we see human competitive (or even better) pattern matching and modeling capabilities. It is not at all trivial to use these capabilities to establish creative tasks, and especially human communication skills are still beyond of what algorithms can do.

Riedl (2016) gives an overview and vision of how AI can approach computational interactive narrative, which requires that computers can understand human communication and react adequately. Attempts in this direction are currently still very limited, as, e.g. shown by Martin, Harrison, and Riedl (2016) in terms of computational improvisation in relatively open (not targeted) communication.

Existing chat bots can answer simple questions in a limited domain of their expertise but lack skills to participate in an open discussion. Recent attempts to improve their capabilities include the ParlAI[14] platform published by Facebook, but these approaches are currently active research directions. Whereas progress in modification of images is astonishing (Zhu, Park, Isola, and Efros (2017) provides a tool that can translate images to other styles, e.g., the style of a specific painter), this is not yet possible for working with text, which is, in this respect, considered much more

[13] `https://botometer.iuni.iu.edu`

[14] https://code.facebook.com/posts/266433647155520/parlai-a-new-software-platform-for-dialog-research/

complex than image translations.

In the related fields of computational creativity and procedural content generation (mainly for games), we see similar problems, which has led to so-called mixed initiative approaches (Liapis, Smith, & Shaker, 2016) where a human designer and a computer program work together, taking turns, in order to reach a specific design goal. Without human interaction, the available methods would not be able to produce results in a human compatible style. At least for the time being, it is seemingly mandatory to employ hybrid approaches in order to establish results that can be taken for generated by a human and thereby appear human-like.

Hybridization as strategy against rule-based detection mechanisms

In order to evaluate our social bots against state-of-the-art detection mechanisms, we confronted them with the BotOrNot service provided by Indiana University (Varol et al., 2017). The BotOrNot service tries to state on the overall probability that a submitted Twitter account is automated. Therefore, the service compares previously learned patterns regarding the account's meta data, network, behavioral timing, friendship relations, sentiment, and content. The authors report of more than $1,150$ features that constitute the patterns in all the named high-level classes. Finally, the results of all indicators are aggregated to a value in $[0, 1]$ which represents a probability of an account being controlled by a social bot. Table 1 shows the overall rating for each continuously active social bot account of our experiment. Obviously, the probability ranges between 0.37 and 0.6 with an average of 0.48. That confirms, in average, no clear bot-identification is possible for our social bots.

In order to judge on the quality of these score distribution for our bots, we generate a baseline distribution of score BotOrNot values of worldwide user accounts.

Methodology. As basis for user extraction we used data from the Twitter Decahose Stream, which provides a random 10% sample of worldwide Twitter traffic. The Twitter Decahose Stream provides roughly 300 posts per second. This sums up to about 160 GB of data per day. From this huge data sample of a single day, we extracted

unique user accounts at four points in time: midnight, morning (6 am), noon (12 am), and evening (6pm) to respect possible effects of the day-night-cycle. The gathered user accounts (about 1200) were classified by using the BotOrNot-API provided by the BotOrNot service. As our Social bots acted in the German language domain, we additionally extracted only German user accounts at the same points in time for a second, localized baseline distribution of scores.

Comparison of Bots and average accounts. The comparison of our social bots' overall scores to the baseline distributions for the worldwide and and German users is shown in Figure 5. Although the bots cannot clearly be classified as bots with respect to the score measure, in retrospective evaluation, their score is significantly higher than the baseline score of our sample score from the worldwide and German Twitter Stream.

To further analyze these findings, we additionally take a look at the detailes meta-features provided by BotOrNot and the according scores.

Content-related features: Figure 6 shows the detailed results for the sentiment, content, and language scores. For the sentiment score features like happiness, valence, arousal, and dominance as well as polarization and emoticon statistics of tweets are evaluated and aggregated. Here our bots obviously behave like baseline German users. Both, German users and bots are generally scored higher than the worldwide baseline, which may be caused by the fact, sentiment analysis for the German language is more difficult than for e.g. English. The same observation can be made for the content feature, which aggregates tweet length and entropy. Here, the bot also range in the German baseline. Language features combine statistics on part-of-speech tags in tweets, i.e. low level features on the tagged or annotated grammar and context of words used in the tweets. Here, a significant difference to both baselines is observed. The reason for this may be the high amount of slang terms and thus grammatically complex structure of tweets used to push a topic in phase 2 of our social bot experiment.

Meta data-related features: For meta data features we observe that our bots behave in average similar to the German user baseline, except for the user score. The user

score aggregates account-specific meta data information like age of the account and profile description as well as frequency and temporal development of actions on Twitter. Especially for these features, our experimental bot accounts are certainly too short-lived to be classified human-like. All other meta-features, however, confirm human-likeness of the social bot behavior—especially, when we consider friendship features, networking and temporal behavior. Here, no significant difference to the German baseline accounts can be identified.

A comment on detection mechanisms and hybridization. The evaluation of our social bot network showed that multiple features of a state-of-the-art detection tool like BotOrNot can be bypassed. Especially the scores "attacked" by our automation framework (friendship, network, temporal behavior) are not distinguishable for bots and the evaluated random German account sample. Only features on content and the user profile showed some indications for bot behavior. These indicators, however, are only identifiable due to an a-priori grouping of the known social bot accounts. If confronted with a single bot account, the BotOrNot detection mechanism does not provide a sufficient overall score to identify any of our social bots as such.

Conclusion

With this paper, we have contributed an interdisciplinary perspective on social bot taxonomy, degrees of automation, developmental costs, and the benefit and importance of human interaction for making social bots invisible for modern detection mechanisms. In detail, we gave a consolidated definition of social bots and applied it to known variants of automated actors in the web. From a more technical perspective, we provided insight into the implementation and costs necessary to deploy simple but reactive social bots in Twitter. To increase credibility, we extended the simple bot implementation by mimicking human behavior in temporal and operational properties. Content production was left to human controllers leading to a hybrid bot network. We experimentally deployed such a network and demonstrated its principle applicability. Tedious tasks were automated (like collecting followers, re-tweeting, or posting

human-prepared content). Finally, we discussed the costs and current technological limits for full human-like hybridization. Futhermore, by means of an empirical analysis from the Twitter bot experiment and average user data extracted from the Twitter Decahose Stream, we have shown that hybrid social bots are able to bypass important indicators of current rule-based detection mechanisms as BotOrNot.

Our results reveal several new challenges for future research in social bot detection: The next big challenge for detection systems will be to identify hybrid social bots, which expose real human behavior, on the one hand, and automatic patterns in some actions, on the other hand. We assume, that rule-based methods will not suffice for these tasks. In fact, adaptive and real-time detection mechanisms, which are able to reconfigure and learn are necessary to react on changing behavioral patterns almost instantly. Additionally, we believe that the inclusion of human interaction into hybrid social bots should shift the focus from purely automatic detection systems to hybrid detection systems that are able to judge on content, background strategies and distributed narratives by the inclusion of human (possibly crowd) intelligence.

Table 1

List of all Twitter bots used during the experiment including the probability of being a Twitter bot determined by BotOrNot for each account.

Bot name	Bot Twitter ID	BotOrNot	Bot name	Bot Twitter ID	BotOrNot
MagaritaWolff	803538518014300160	40%	44Maler	803586351807479809	40%
KumlehnLisa	803584267653709824	39%	FlorianWanken	803586889802465280	47%
DreysKatharina	803580625911480320	49%	IHeulach	803917039597473792	47%
LaurySamsy	803575433862283264	53%	porryflo12	803915544755847168	48%
Eva_Omaha	803577951040180224	46%	DiamondGirl_97	803595282969755648	40%
Jonas_Der_Baum	803581393393643520	37%	kenny_boy300	803278016709332994	42%
NickyTheMan1	803580666260705281	42%	Saschamachtsgut	803279247804628992	41%
ruediwig	803579187344904192	56%	ollerbaum121	803574359281598464	52%
Kalle_dod	803271431400394752	54%	DinoDingi	803582934376738816	60%
Shagggy_93	803274850768920576	46%	wernerbbbright	803584898271444992	48%
Luise_D2	803583207069351936	58%	The_pfist	803881433840361472	47%
hoppendorf	803584623594897409	51%	hansemeister11	803901587827621889	54%
sabinepeterson7	803585050721783808	57%	wendtneraxxxy	803912137630420992	56%
ullaschoene80	803988474416295937	54%			

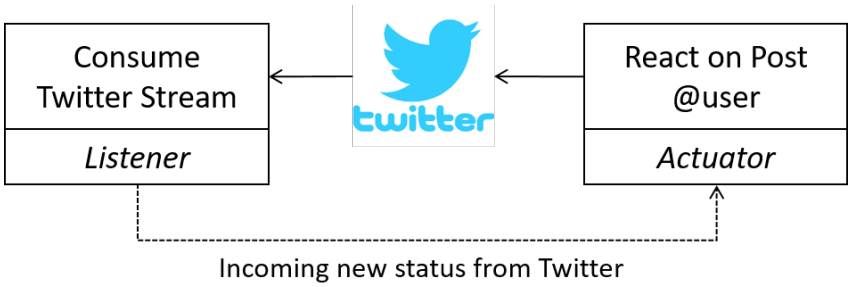

Figure 1. Components and data flow of simple Twitter bot realization using the Twitter Stream API.

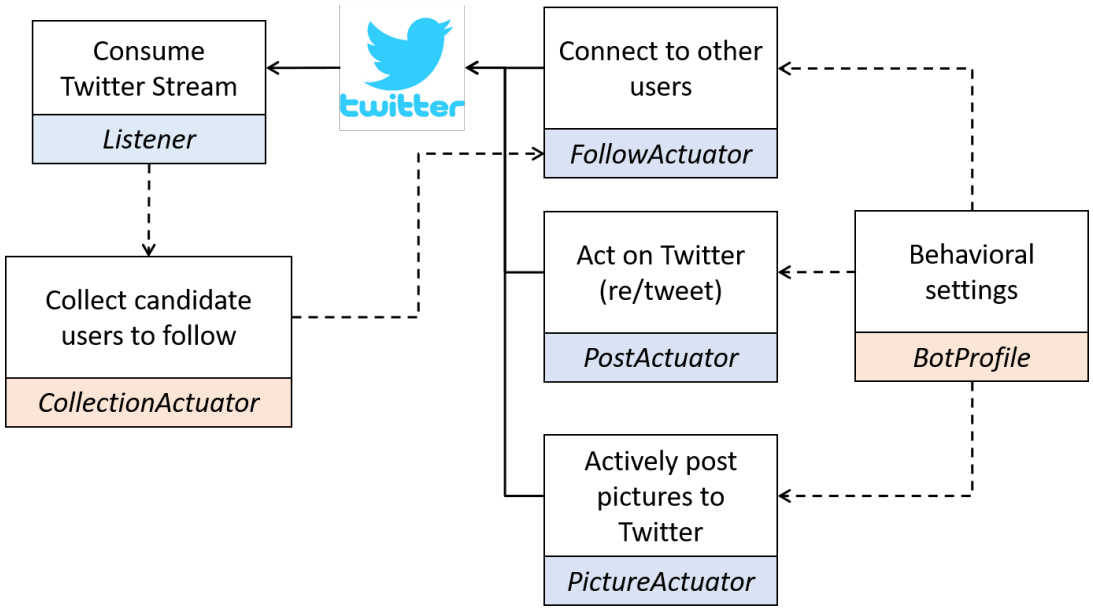

Figure 2. Components and data flow of our advanced social bot with behavioral settings, follow-for-follow mechanism, and human-like activity profile.

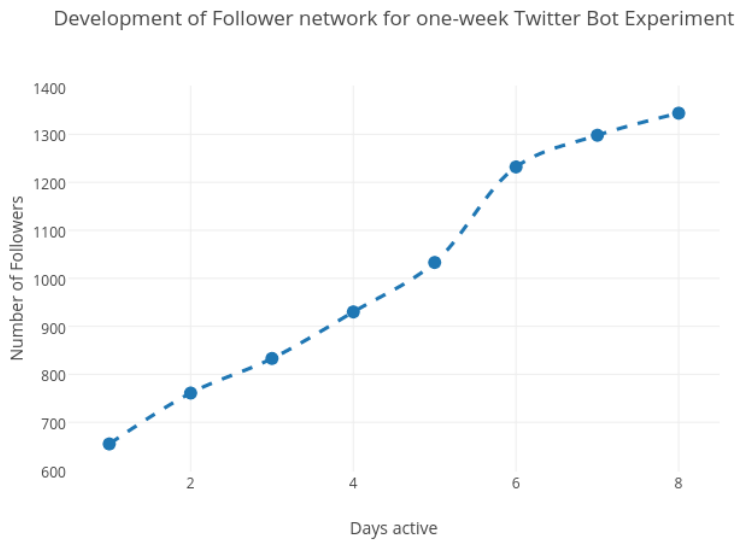

Figure 3. The plot shows the growth of the follower network for the initial Twitter bot setup in about one week. 27 of initially 30 Twitter bots continuously performed the follow-for-follow strategy automatically without any human intervention. Potential followers were selected from the Twitter stream regarding individual topics.

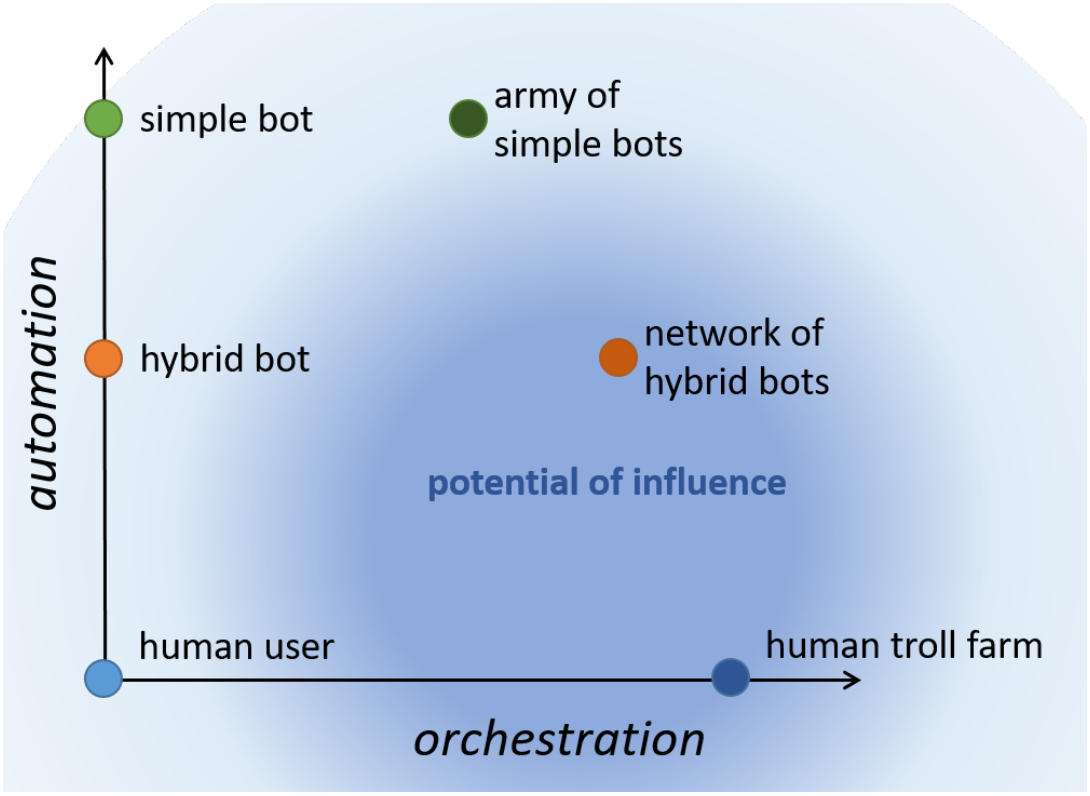

Figure 4. Qualitative classification of the potential influence of humans and bots in social media with respect to automation and orchestration.

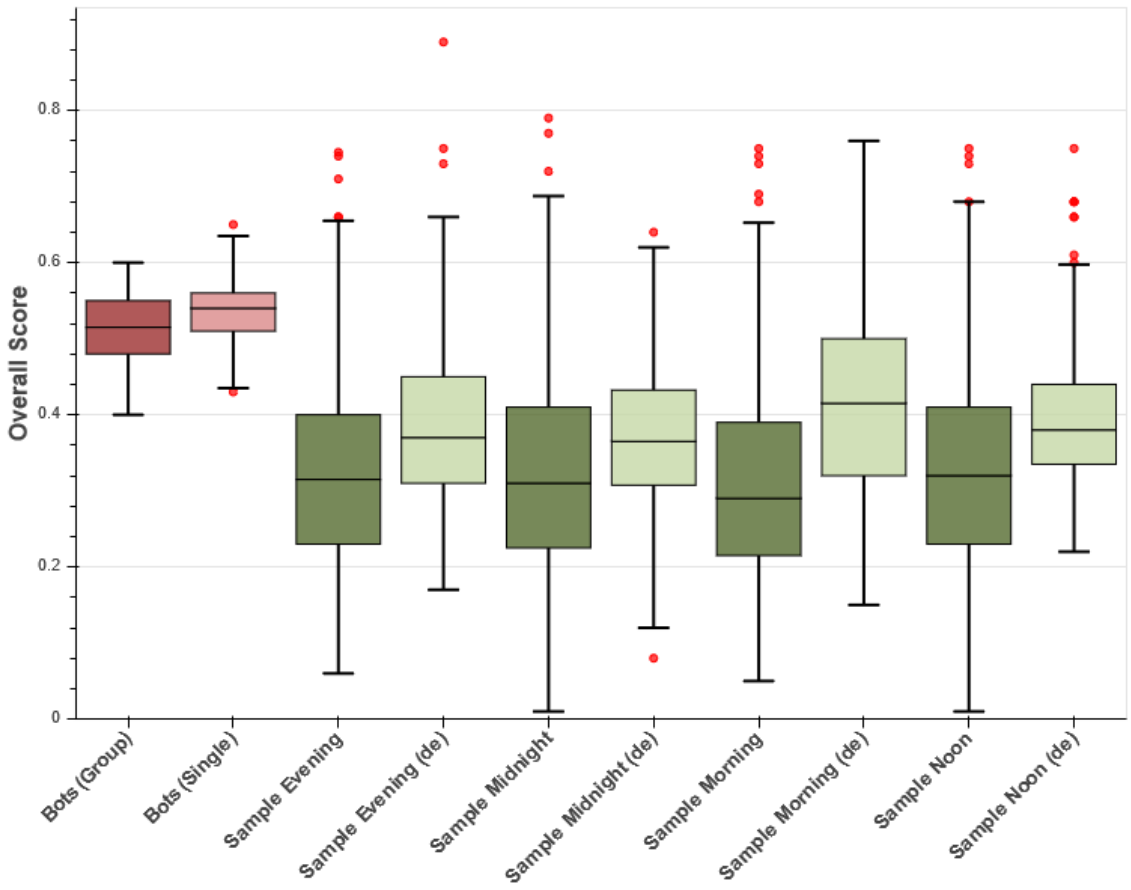

Figure 5. Statistics of the overall BotOrNot score values for our the social bot network (red box plots, grouped into bots that initially act as single entity or group respectively) contrasted with two baseline overall scores for a set of sample users. The sample users are taken from the worldwide (green box plots) and German (light green box plots) Decahose Twitter stream at four points in time. The analysis was performed using the BotOrNot API.

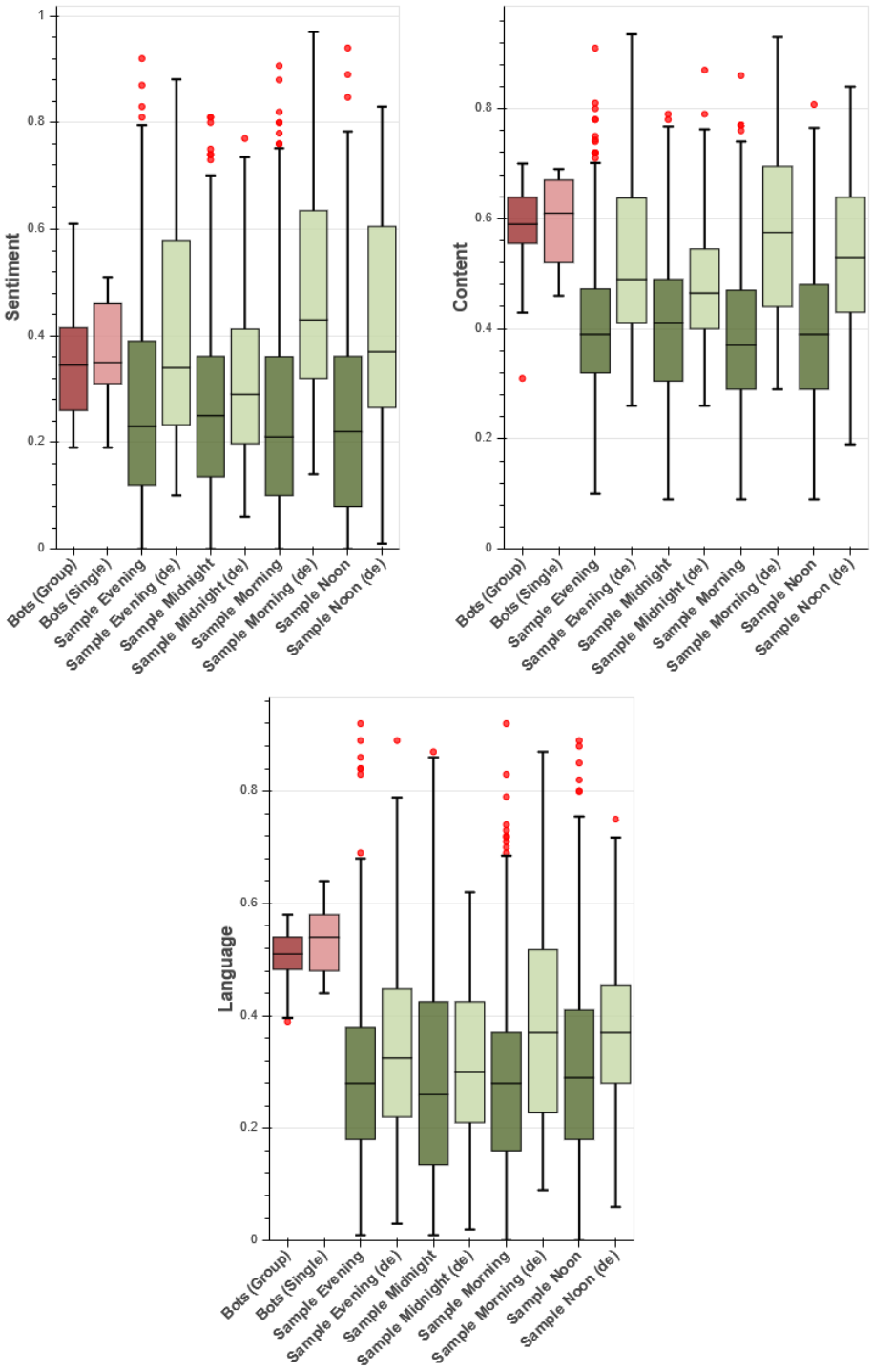

Figure 6. Detailed statistics of three meta features (sentiment, content, and language) for our social bots (red) and the baseline accounts worldwide (green) and from Germany (light green).

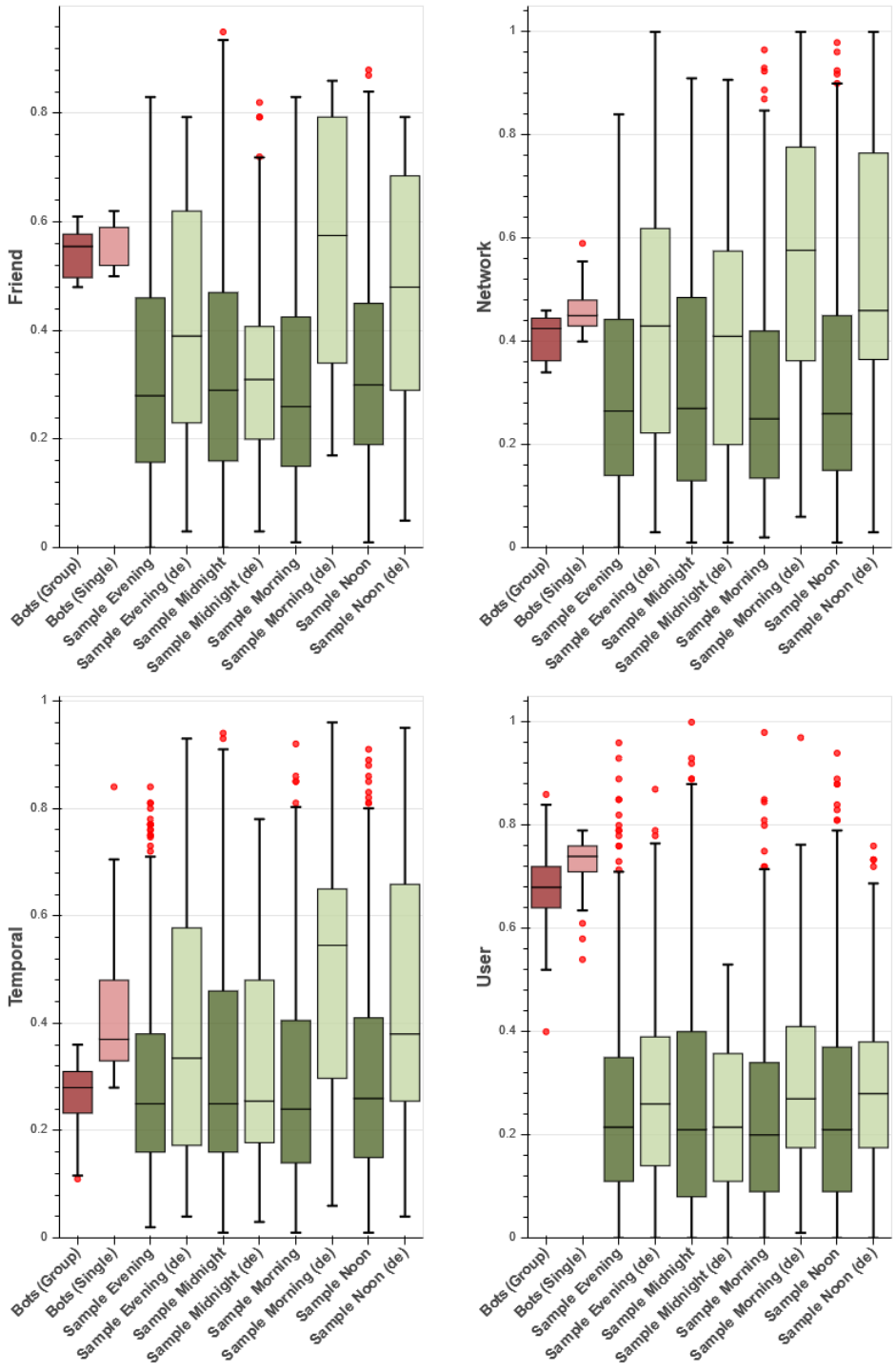

Figure 7. Detailed statistics of four meta features (friendship, network, temporal, and user) for our social bots (red) and the baseline accounts worldwide (green) and from Germany (light green).

References

Bessi, A., & Ferrara, E. (2016). Social bots distort the 2016 us presidential election online discussion. *First Monday*, *21*(11).

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2010). Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Proceedings of the 26th annual computer security applications conference* (pp. 21–30). New York, NY, USA: ACM. doi: 10.1145/1920261.1920265

Cordy, J. (2017). *The social media revolution: Political and security implications* (Draft CDS Report No. [064 CDSDG 17 E]). NATO Parliamentary Assembly, Sub-Committee on Democratic Governance. Retrieved from `http://www.nato-pa.int`

Elliott, C. (2014, Mai). *The readers' editor on... pro-Russia trolling below the line on Ukraine stories.* online. (http://www.theguardian.com/commentisfree/2014/may/04/pro-russia-trolls-ukraine-guardian-online)

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Commun. ACM*, *59*(7), 96–104.

Fredheim, R. (2013). *Putin's bot army – part one: a bit about bots.* online. (http://quantifyingmemory.blogspot.co.uk/2013/06/putins-bots-part-one-bit-about-bots.html)

Frischlich, L., Boberg, S., & Quandt, T. (2017). Online hate speech. In K. Kaspar, L. Gräßer, & A. Riffi (Eds.), (chap. Un-menschlicher Hass: Die Rolle von Empfehlungsalgorithmen und Social Bots für die Verbreitung von Cyberhate). kopaed.

Geiger, R. S. (2016). Bot-based collective blocklists in twitter: the counterpublic moderation of harassment in a networked public space. *Information, Communication & Society*, *19*(6), 787–803. Retrieved from

http://dx.doi.org/10.1080/1369118X.2016.1153700

Hegelich, S. (2016). Invasion der Meinungs-Roboter. *Analysen & Argumente*, *221*(2016), 1–9.

Heinrich-Böll-Stiftung. (2017, 02). *Social bots.* https://www.boell.de/de/2017/02/09/social-bots.

Kollanyi, B., Howard, P. N., & Woolley, S. C. (2016). *Bots and automation over twitter during the u.s. election* (Tech. Rep. No. Data Memo 2016.4). www.politicalbots.org: Oxford, UK: Project on Computational Propaganda.

Lecun, Y., Bengio, Y., & Hinton, G. (2015, 5). Deep learning. *Nature*, *521*(7553), 436–444. doi: 10.1038/nature14539

Lehmann, S. (2013, 12). *You are here because of a robot.* https://sunelehmann.com/2013/12/04/youre-here-because-of-a-robot/.

Liapis, A., Smith, G., & Shaker, N. (2016). Mixed-initiative content creation. In *Procedural content generation in games* (pp. 195–214). Cham: Springer International Publishing. doi: 10.1007/978-3-319-42716-4_11

Maréchal, N. (2016). Automation, algorithms, and politics| when bots tweet: Toward a normative framework for bots on social networking sites (feature). *International Journal of Communication*, *10*(0).

Martin, L. J., Harrison, B., & Riedl, M. O. (2016). Improvisational computational storytelling in open worlds. In F. Nack & A. S. Gordon (Eds.), *Interactive storytelling: 9th international conference on interactive digital storytelling, icids 2016, los angeles, ca, usa, november 15–18, 2016, proceedings* (pp. 73–84). Cham: Springer International Publishing. doi: 10.1007/978-3-319-48279-8_7

Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *Aaai* (Vol. 94, pp. 16–21).

Mitterhofer, S., Kruegel, C., Kirda, E., & Platzer, C. (2009). Server-side bot detection in massively multiplayer online games. *IEEE Security & Privacy*, *7*(3).

Pfaffenzeller, M. (2017, Mar). *Bundestagswahlkampf: CDU erwägt Einsatz von Chatbots.* SPIEGEL ONLINE. Retrieved from

http://www.spiegel.de/netzwelt/web/cdu-peter-tauber-erwaegt-einsatz-von-chatbots-im-bundestagswahlkampf-a-1141207.html

Riedl, M. O. (2016). Computational narrative intelligence: A human-centered goal for artificial intelligence. *CoRR*, *abs/1602.06484*. Retrieved from http://arxiv.org/abs/1602.06484

Rosenbach, S., Marcel. (2016, Oct). *Internet-Kommentare von Automaten: AfD will im Wahlkampf Meinungsroboter einsetzen.* SPIEGEL ONLINE. Retrieved from http://www.spiegel.de/netzwelt/netzpolitik/afd-will-im-wahlkampf-social-bots-einsetzen-a-1117707.html

Shawar, B. A., & Atwell, E. (2007a). Chatbots: are they really useful? In *Ldv forum* (Vol. 22, pp. 29–49).

Shawar, B. A., & Atwell, E. (2007b). Different measurements metrics to evaluate a chatbot system. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies* (pp. 89–96).

Tynan, D. (2012, Apr). *Social Spam is taking over the Internet.* Retrieved from http://www.itworld.com/article/2832566/it-management/social-spam-is-taking-over-the-internet.html

Varol, O., Ferrara, E., Davis, C., Menczer, F., & Flammini, A. (2017). *Online human-bot interactions: Detection, estimation, and characterization.* Retrieved from https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15587

Weedon, J., Nuland, W., & Stamos, A. (2017, 04). *Information operations on facebook* (Tech. Rep. No. 1). Facebook, Inc.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Woolley, S. (2016). Automating power: Social bot interference in global politics. *First Monday*, *21*(4).

Zhu, J., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, *abs/1703.10593*. Retrieved

from http://arxiv.org/abs/1703.10593

Appendix

Code of a Simple Twitter Bot

The following code is a fully functional Twitter bot which continuously tracks the Twitter stream for a given hash tag (`#Hashtag`) and instantly replies to the sender with a simple 'Hello'. Please note, that login information for the Twitter API has been obscured and some error handling code has been removed for brevity.

```
import tweepy
from tweepy.parsers import RawParser

# followed topic and identity of bot
search_topics = ['#Hashtag']
bot_identity = 'twitter_name'

# API object
api = None

# Implements simple bot actions
class TwitterActuator:

    def act(self, status):
        if not api is None:
            username = status.author.screen_name
            tweettext = status.text
            if not username == bot_identity:
                print('Received:' + tweettext + '\n')
                # tweet to the world
                api.update_status('Hello @' + username + '.')

# Stream Listener (reacts on twitter posts)
class TwitterStreamListener(tweepy.StreamListener):
    actuator = None

    # set the actuator
    def setActuator(self, actuator):
        self.actuator = actuator

    # Call the actuator if posts appear in stream
    def on_status(self, status):
        self.actuator.act(status)

# Main Program
```

```
if __name__ == '__main__':
    try:
        auth = tweepy.OAuthHandler('***', '***')
        auth.set_access_token('***', '***')

        api = tweepy.API(auth_handler=auth,
                         parser=RawParser(), wait_on_rate_limit=True)

        twitterBot = TwitterActuator()
        twitterListener = TwitterStreamListener()
        twitterListener.setActuator(twitterBot)

        myStream = tweepy.Stream(auth = api.auth,
                         listener=twitterListener)

        myStream.filter(track=search_topics, languages=['en'],
                         encoding='utf-8')

    finally:
        print('Program end.')
```